

• Programación estructurada

.Expresiones aritméticas y apuntadores

- Con los apuntadores se pueden realizar un conjunto limitado de operaciones aritméticas

- Debido a que los apuntadores hacen referencia a direcciones de memoria solo pocas operaciones aritméticas tienen congruencia.
- Un apuntador puede ser incrementado, decrementado o sumado, sustraído de otro apuntador.

•ejemplo

```
-int v[10];
```

.Suponga que una variable entera ocupa 4 bytes.

.Si.

```
-vPtr = &v[0];
```

.o también

```
-vPtr = v;
```

.entonces

```
-vPtr = 3000
```

```
-vPtr + 2 =
```

.3008

•ejemplo



•vPtr = 3000

•vPtr + 2 = 3008

•Evaluación

.Si $v[] = \{4, 8, 6, 7, 9, 5, 4, 2, 3, 9\}$;

.y $\&v[0] = 4008$

.¿Cuanto es?

–v

–vPtr

–*vPtr

–&v

–&*vPtr

–vPtr + 3

–*(vPtr + 1)

–*vPtr + 5

–*(v + 2)

```

#include <stdio.h>
#include <conio.h>

main()
{
    int i,b[]={10,20,30,40};
    int *bPtr=b;

    printf("Arreglo b impreso con la notacion:\n"
           "\tArreglo suindice\n");
    for(i=0;i<=3;i++)
        printf("\tb[%d] = %d\n",i,b[i]);

    printf("\tnombre del arreglo subindice\n");
    for(i=0;i<=3;i++)
        printf("\t*(b + %d) = %d\n",i,*(b+i));

    printf("\tapuntador al arreglo subindice\n");
    for(i=0;i<=3;i++)
        printf("\t*(bPtr + %d) = %d\n",i,*(bPtr+i));

    getch();
    return 0;
}

```